Master's Thesis

An Improved Low-Rank Decomposition Model with Total Variation Regularization for Detecting Foreground

Guidance

Professor Nobuo YAMASHITA
Program-Specific Associate Professor Hiroyuki SATO

Reiichi SUGAO

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2019

Abstract

In this paper we consider problems of decomposing a given matrix into low-rank and sparse components, which appear in various fields such as pattern recognition, image denoising and video surveillance. To get the decomposition we usually solve a minimization problem of two matrix norms: a nuclear norm and an L1 norm, which induce low-rank structures and sparsity, respectively. Recently, Xiaochun et al. proposed a new minimization model with a total variation (TV) regularization term in addition to the two norms, and presented an alternating direction method of multipliers (ADMM) with multi-block variables to solve it. The TV terms are represented as a sum of the difference between adjacent components in the matrices. When the minimization model is applied to foreground detection, the TV terms enable us to detect spatially and temporally continuous foreground. However, since Xiaochun et al.'s TV terms include the difference of the terminal components in the matrices, they regard the first and last columns in the matrices as adjacent ones. Moreover, the multi-block ADMM has redundant computations, and is not necessarily globally convergent.

To overcome these difficulties, we first propose an improved TV model without calculating the difference between the terminal components. Then, we show that the existing simple ADMM can solve minimization problems with TV terms by elaborating two blocks in the ADMM. Finally, we conduct numerical experiments for foreground detection. The numerical results show that the proposed method tends to obtain more accurate solutions compared with the original model, particularly with respect to the terminal components in the matrices. The proposed model also is more efficient, at least with the instances considered in this work.